Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Волгоградский государственный технический университет»

Факультет электроники и вычислительной техники

Кафедра «Программное обеспечение автоматизированных систем»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовому проекту**

по дисциплине «Междисциплинарный курсовой проект»

на тему: «Разработка программно-информационной системы»

Группа: ПрИн-466, ПрИн-467

Проект зачтен с оценкой:

Носкин В. В. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

Дудкин Д. М. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

Крымова М. В. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

Руководитель проекта, нормоконтроллер \_\_\_\_\_\_\_\_\_\_\_\_ Литовкин Д.В.

Члены комиссии:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_               Ю.А. Орлова

(подпись и дата подписания) (инициалы и фамилия)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_                 Е.Е. Громов

(подпись и дата подписания) (инициалы и фамилия)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_                 А.В. Аникин

(подпись и дата подписания) (инициалы и фамилия)

Волгоград 2020 г.

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Волгоградский государственный технический университет»

Факультет электроники и вычислительной техники

Направление «09.03.04 Программная инженерия»   
Кафедра «Программное обеспечение автоматизированных систем»

Дисциплина «Междисциплинарный курсовой проект»

Утверждаю

Зав. кафедрой \_\_\_\_\_\_\_\_\_\_ Орлова Ю.А.

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_ г.

**ЗАДАНИЕ**

**на курсовой проект**

Студенты: Носкин В. В., Дудкин Д. М., Крымова М. В.

Группа: ПрИн-466, ПрИн-467

1. Тема: «Разработка программно-информационной системы»

Утверждена приказом №    892-ст от « 31 »    августа    2020 г.

2. Срок представления проекта к защите «  25  » декабря 2020 г.

3. Содержание расчетно-пояснительной записки:

1) формулировка задания; 2) спецификация требований; 3) модель данных; 4) используемые инструменты; 5) маршрутизация; 6) репозиторий кода; 7) тестирование; 8) полученное программное решение

4. Дата выдачи задания «  17  »   сентября   2020 г.

Руководитель проекта: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Литовкин Д.В.

Задание принял к исполнению: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Носкин В. В.

Задание принял к исполнению: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Дудкин Д. М.

Задание приняла к исполнению: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Крымова М. В.

Содержание

[1 Формулировка задания 4](#_Toc62403831)

[2 Спецификация требований к программному обеспечению 4](#_Toc62403832)

[2.1 Сценарии использования 4](#_Toc62403833)

[2.2 Макеты экранных форм 8](#_Toc62403834)

[2.3 Диаграммы состояния экранных форм 12](#_Toc62403835)

[3 Модель данных 14](#_Toc62403836)

[4 Используемые инструменты 14](#_Toc62403837)

[5 Маршрутизация 15](#_Toc62403838)

[6 Репозиторий кода 18](#_Toc62403839)

[7 Тестирование 18](#_Toc62403840)

[8 Полученное программное решение 18](#_Toc62403841)

[9 Вклад участников команды 18](#_Toc62403842)

[10 Используемые источники 20](#_Toc62403843)

# 1 Формулировка задания

Учет групп студентов в деканате. Цель - получить список студентов заданной группы на указанную дату.

Ключевые варианты использования программы:

* Зачисление студентов на первый курс бакалавриата или магистратуры;
* перевод студентов на старший курс;
* отчисление студентов в связи с окончанием обучения;
* получение списка студентов заданной группы на указанную дату.

Используются следующие допущения:

* В период обучения студенты НЕ отчисляются;
* переводы студентов НЕ осуществляются;
* зачисление студентов из других вузов НЕ осуществляется.

# 2 Спецификация требований к программному обеспечению

## 2.1 Сценарии использования

Сценарий “Просмотр списка групп”:

1. Пользователь выбирает учебный год (по умолчанию самый поздний);
2. Пользователь задает курс;
3. Пользователь задает направление;
4. Система отображает список групп с пометкой об их состоянии на выбранный год (отчислена, переведена).

Альтернативные сценарии:

* Для выбранных параметров не найдено групп:

1. выводим сообщение об отсутствии групп.

Сценарий “Добавление группы”:

1. Пользователь выбирает учебный год (по умолчанию самый поздний);
2. Пользователь нажимает кнопку “добавить группу” на странице со списком групп;
3. Система отображает форму для ввода инфо о новой группе;
4. Пользователь задает курс;
5. Пользователь задает направление;
6. Система создает группу с выбранными параметрами в текущем учебном году;
7. Система отображает страницу со списком студентов созданной группы (пока пустым).

Альтернативные:

* параметры группы совпадают с параметрами другой группы (конфликт при переводе)
  1. создаем одинаковые, но у них разные id.

Сценарий “Отчисление всей группы по окончании обучения”:

1. Пользователь выбирает учебный год (по умолчанию самый поздний);
2. Пользователь отмечает группу, студентов которой нужно отчислить;
3. Пользователь выбирает причину отчисления (“окончание обучения”) во всплывающем окне;
4. Система отчисляет каждого студента выбранной группы;
5. Система обновляет статус группы для текущего года;
6. Система отображает обновленный список групп для текущего года.

Альтернативные:

* группа, которую хотим отчислить, уже переведена или отчислена
  1. ничего не делаем, выводим ошибку.
* некоторые студенты уже отчислены
  1. отчислять только студентов, числящихся в группе на данный момент.
* в группе нет студентов
  1. "отчисляем” как обычно, но для 0 студентов.

Сценарий “Просмотр студентов группы”:

1. Пользователь выбирает учебный год (по умолчанию самый поздний);
2. Пользователь кликает по строке группы для просмотра;
3. Система отображает страницу со списком студентов группы.

Альтернативные:

* В группе нет студентов

1. выводим сообщение об отсутствии студентов.

Сценарий “Отчисление одного или нескольких студентов (НЕ всей группы)”:

1. Пользователь выбирает учебный год (по умолчанию самый поздний);
2. Пользователь открывает список студентов группы;
3. Пользователь выбирает одного или нескольких студентов для отчисления;
4. Пользователь нажимает на кнопку “отчислить”;
5. Пользователь выбирает причину отчисления во всплывающем окне;
6. Система создает запись об отчислении студента(-ов);
7. Система отображает обновленный список группы с пометками об отчислении.

Альтернативные:

* Студент, которого хотим отчислить (хотя бы один из выбранных студентов), уже отчислен
  1. ничего не делаем, выводим ошибку.
* Выбранные студенты должны быть отчислены по разным причинам:
  1. пользователь должен отменить действие и выбрать отчисляемых по одной причине.

Сценарий “Зачисление нового студента в группу”:

1. Пользователь выбирает учебный год (по умолчанию самый поздний);
2. Пользователь открывает список студентов группы;
3. Пользователь нажимает на кнопку “зачислить студента”;
4. Система отображает форму для ввода инфо о новом студенте;
5. Пользователь задает ФИО;
6. Система создает запись студента и связывает ее с выбранной группой;
7. Система отображает обновленный список группы.

Альтернативные:

* В системе уже числится студент с таким фио (считаем, что фио уникально)
  1. ничего не делать, вывести ошибку.

Сценарий “Зачисление ранее отчисленного студента в группу”:

1. Пользователь выбирает учебный год (по умолчанию самый поздний);
2. Пользователь открывает список студентов группы;
3. Пользователь нажимает на кнопку “зачислить студента”;
4. Пользователь выбирает опцию “выбрать существующего”;
5. Система отображает список студентов, доступных для зачисления;
6. Пользователь выбирает студента из списка;
7. Система связывает запись выбранного студента с группой;
8. Система отображает обновленный список группы.

Сценарий “Открытие нового учебного года”:

1. Пользователь нажимает “начать новый учебный год”;
2. Система создает пустой учебный год с номером на 1 больше последнего добавленного.

## 2.2 Макеты экранных форм

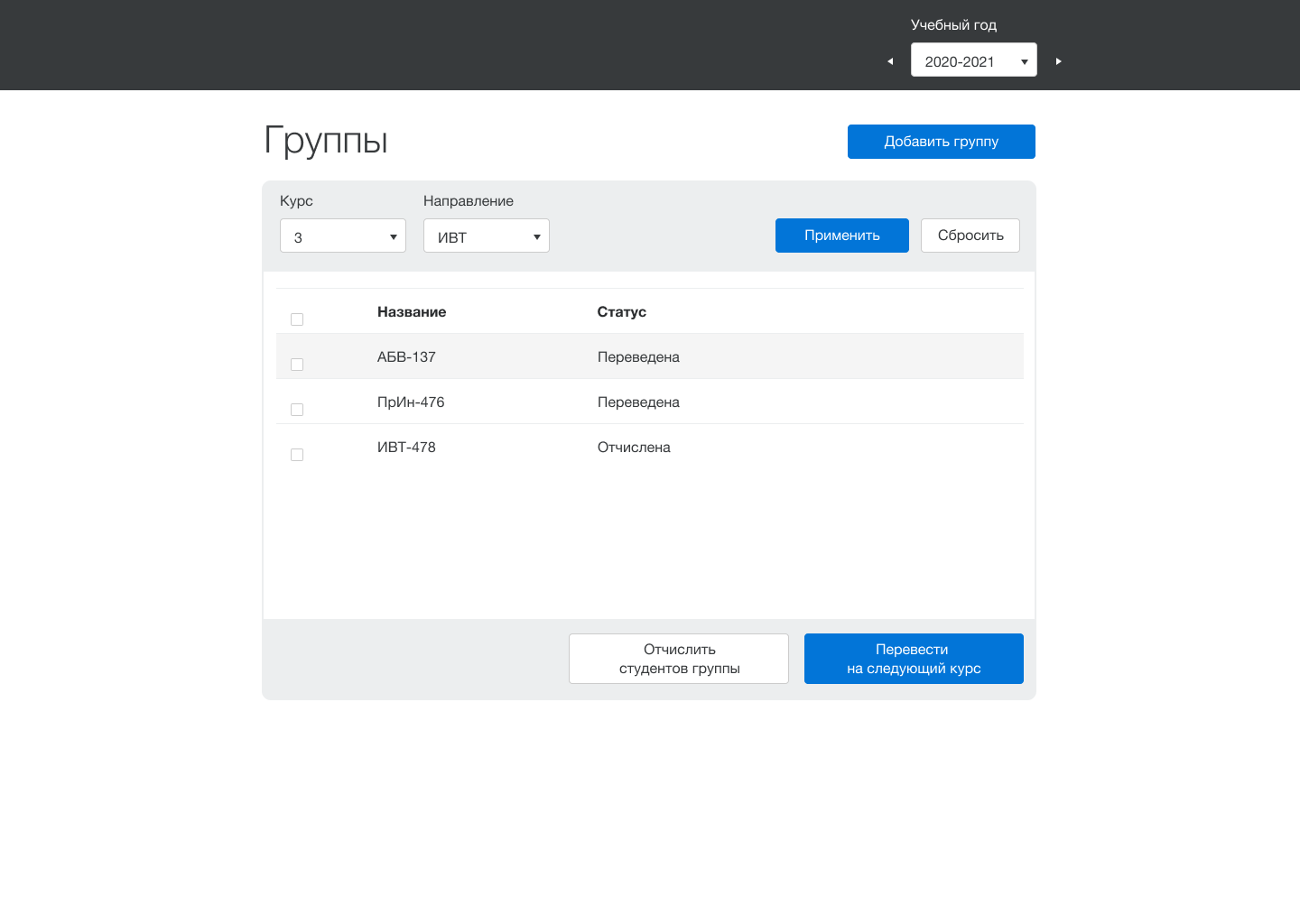


Рисунок 1 – Просмотр списка групп для выбранного года

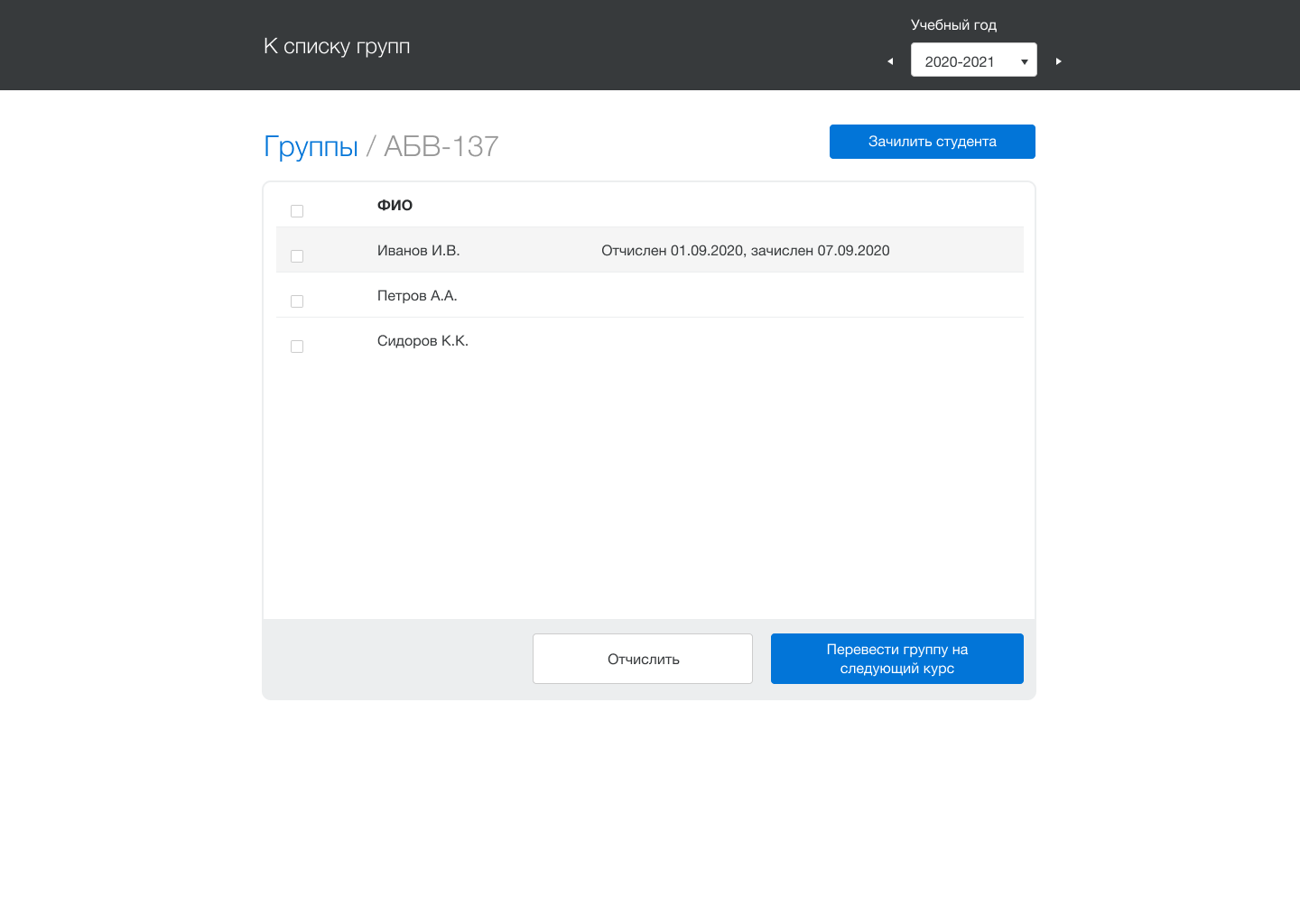


Рисунок 2 – Просмотр состава группы в выбранном году

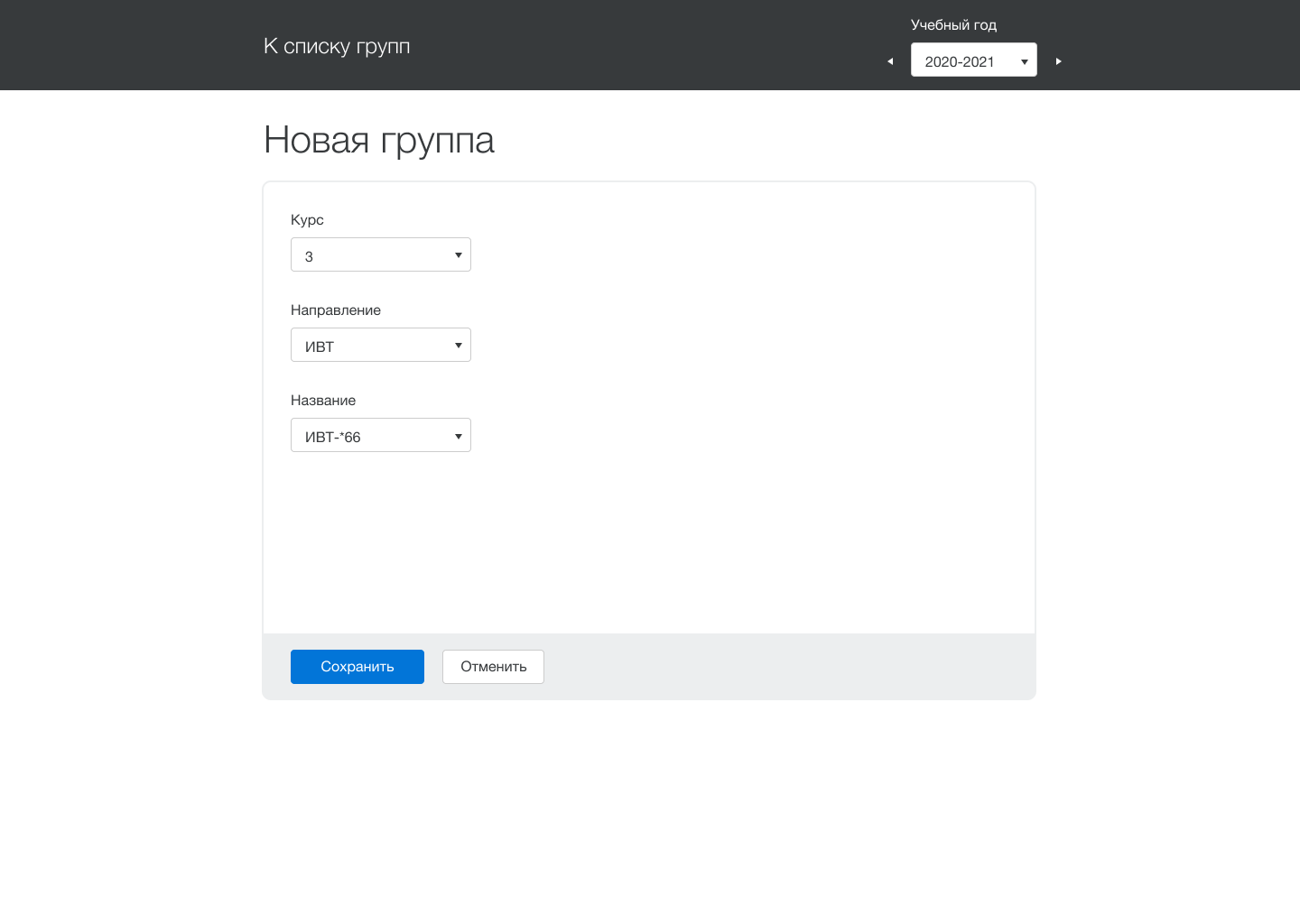


Рисунок 3 – Создание новой группы

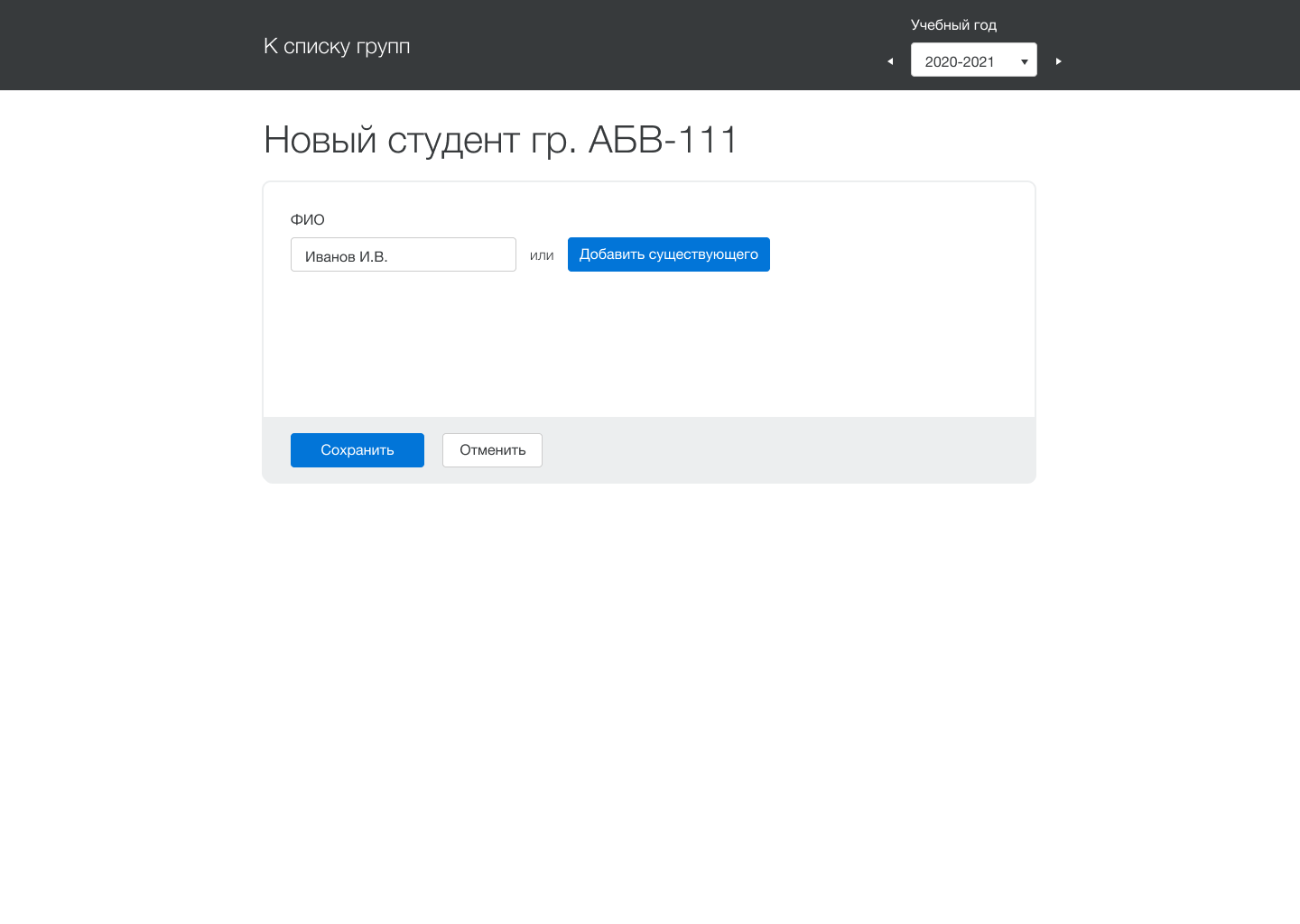


Рисунок 4 – Создание нового студента в группе

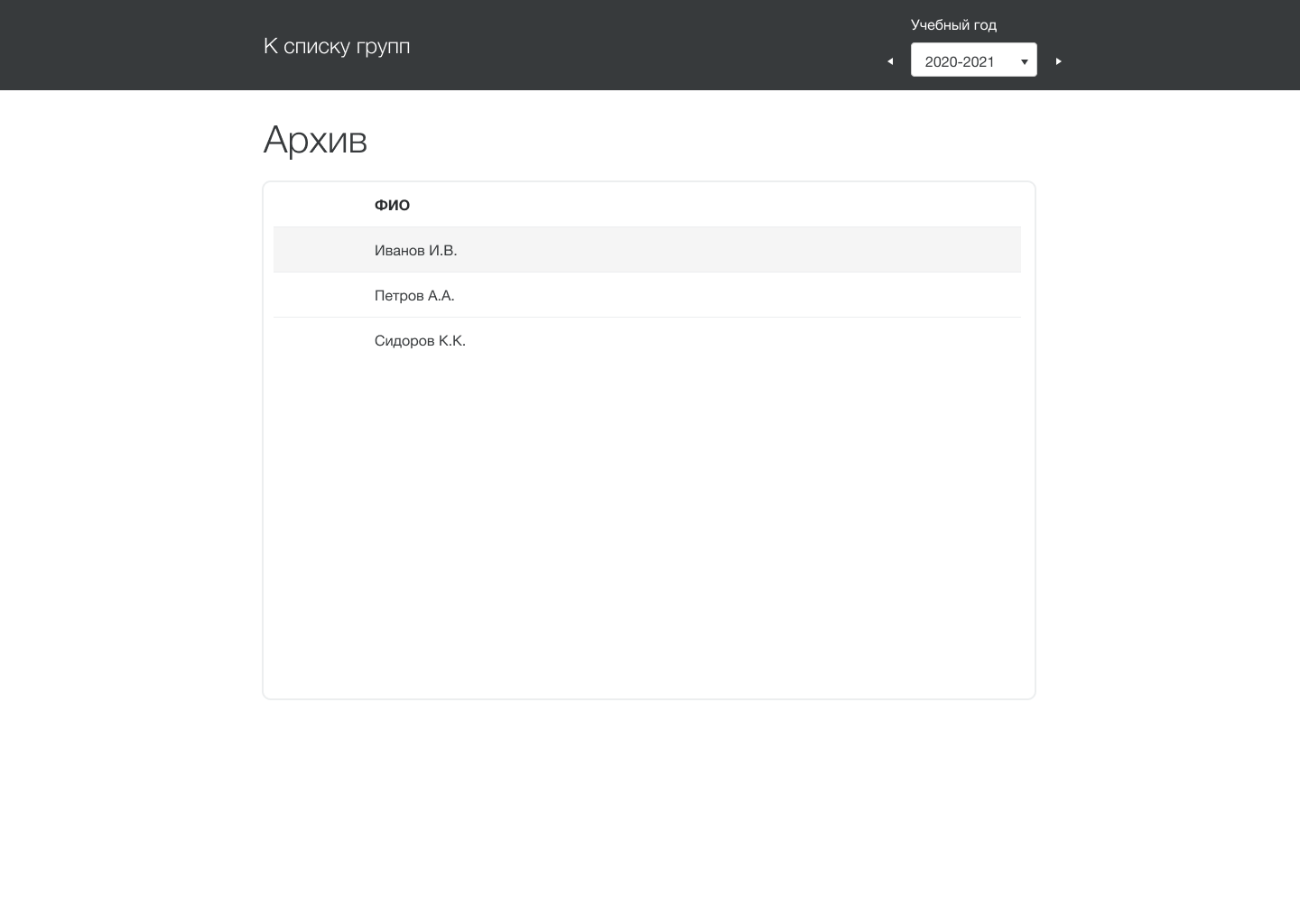


Рисунок 5 – Архив

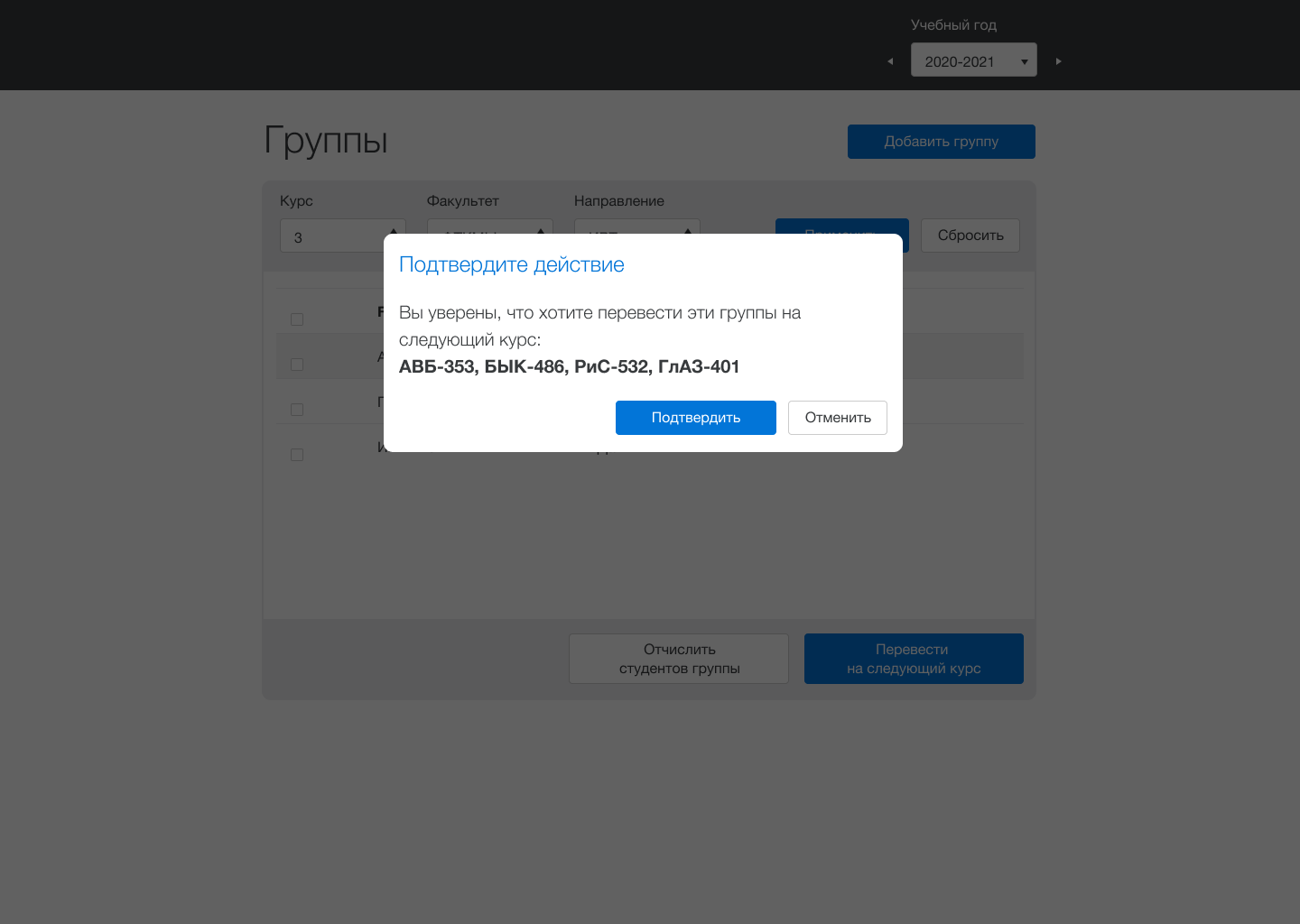


Рисунок 6 – Модальное окно подтверждение о переводе группы

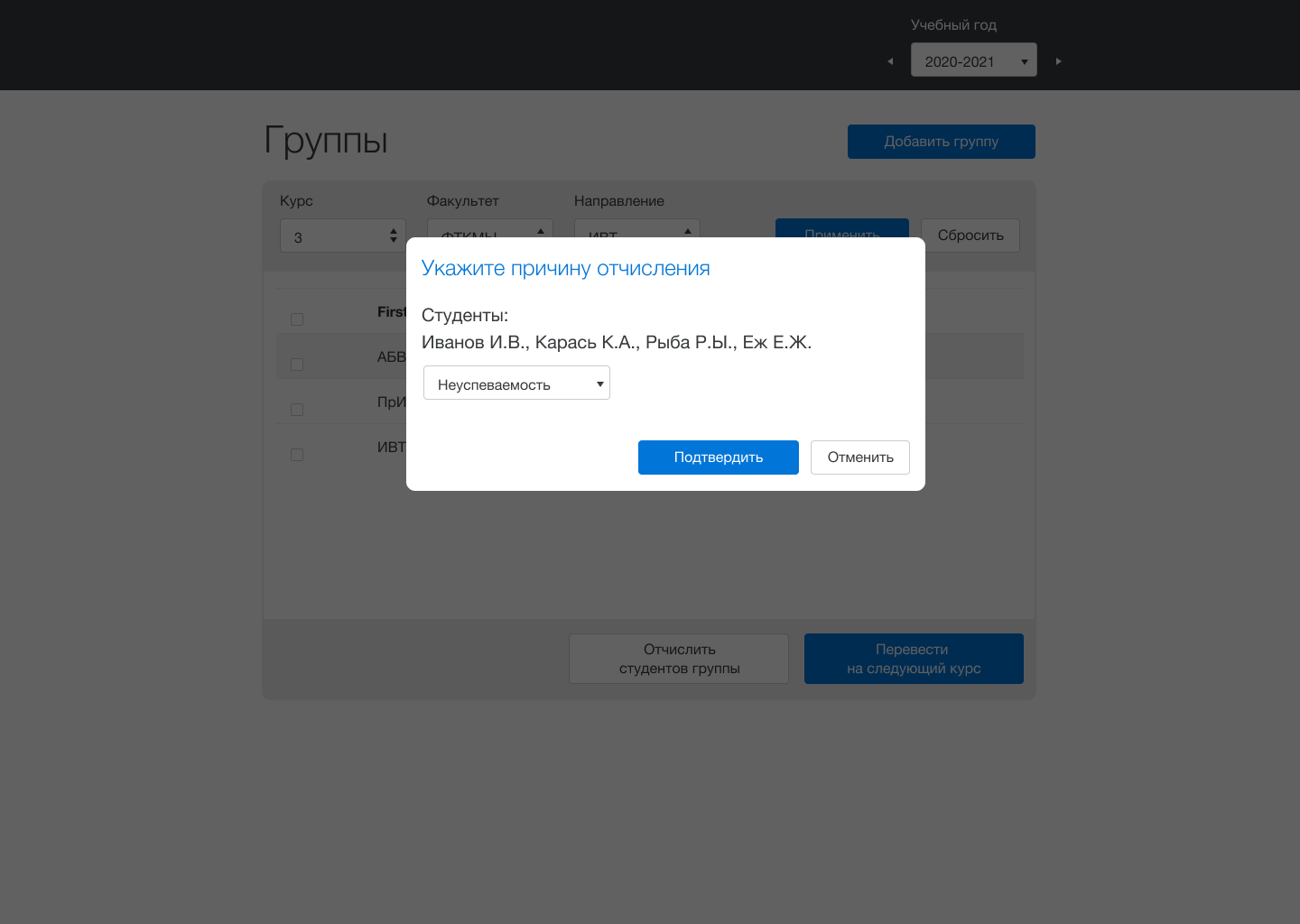


Рисунок 7 – Модальное окно отчисления группы

## 2.3 Диаграммы состояния экранных форм

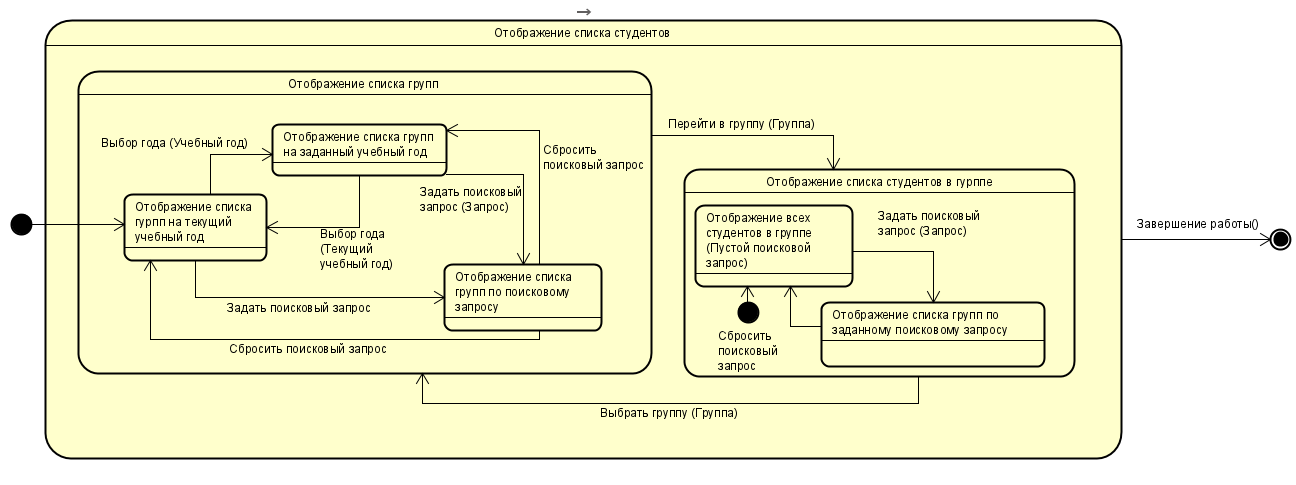


Рисунок 8 – Диаграмма состояния “Просмотр списка групп и студентов”

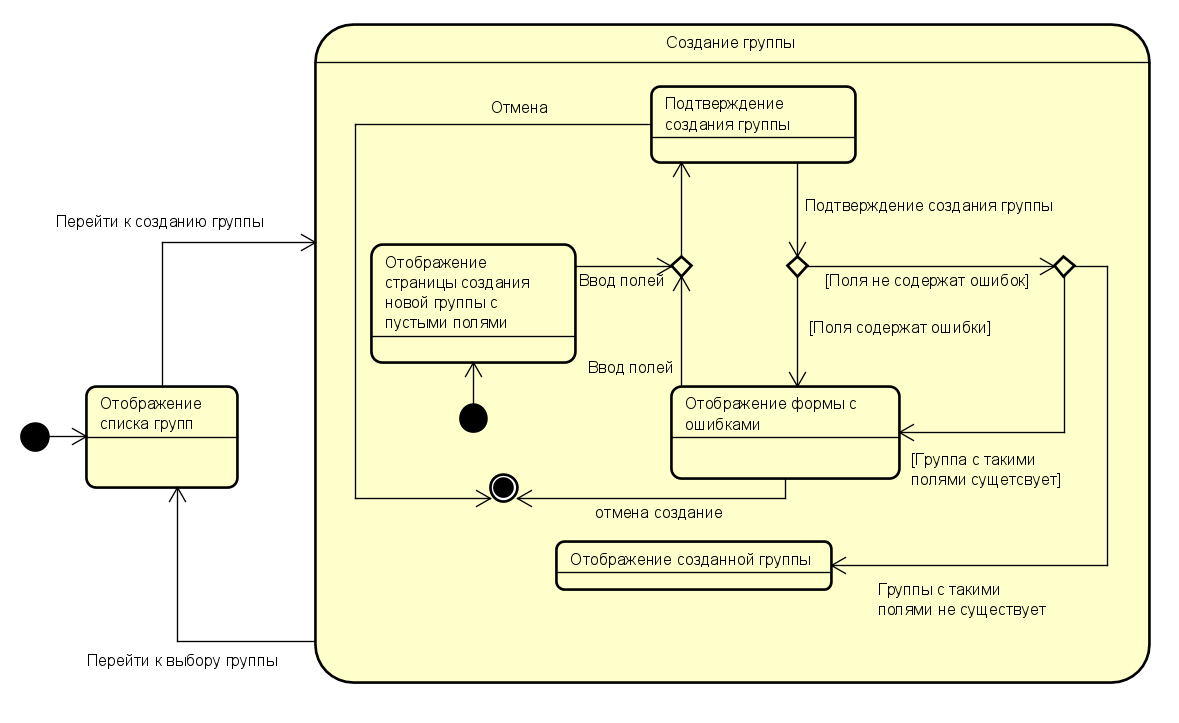


Рисунок 9 – Диаграмма состояния “Создание группы”

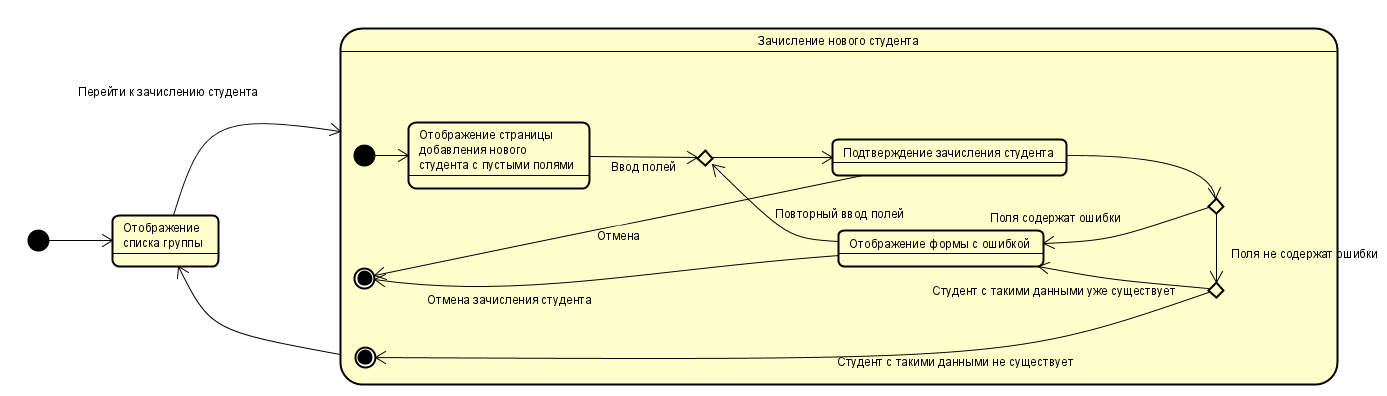


Рисунок 10 – Диаграмма состояния “Зачисление студента”

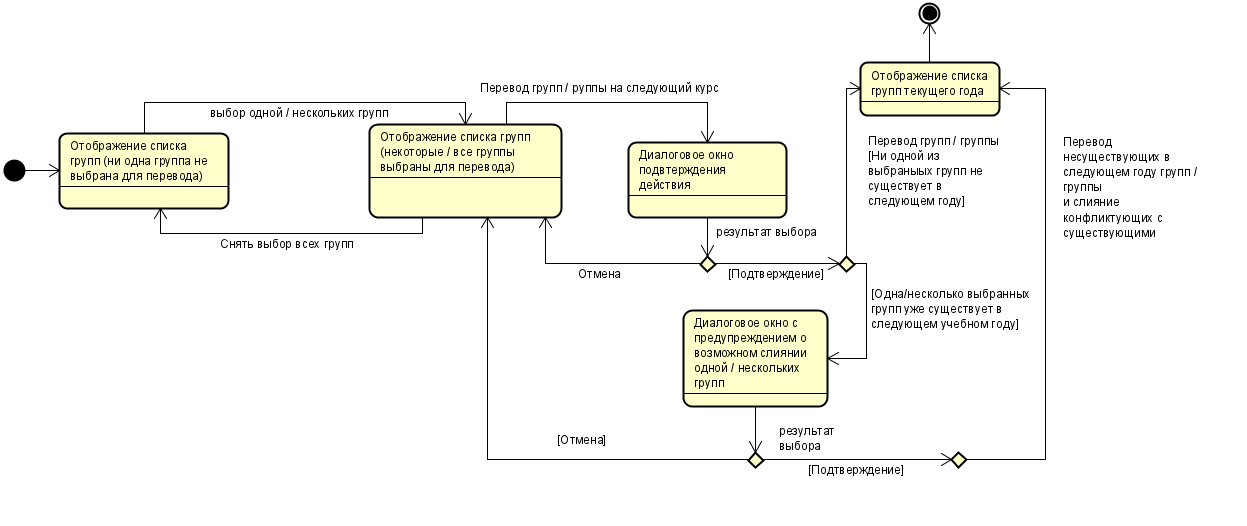


Рисунок 11 – Диаграмма состояния “Перевод группы на следующий курс”

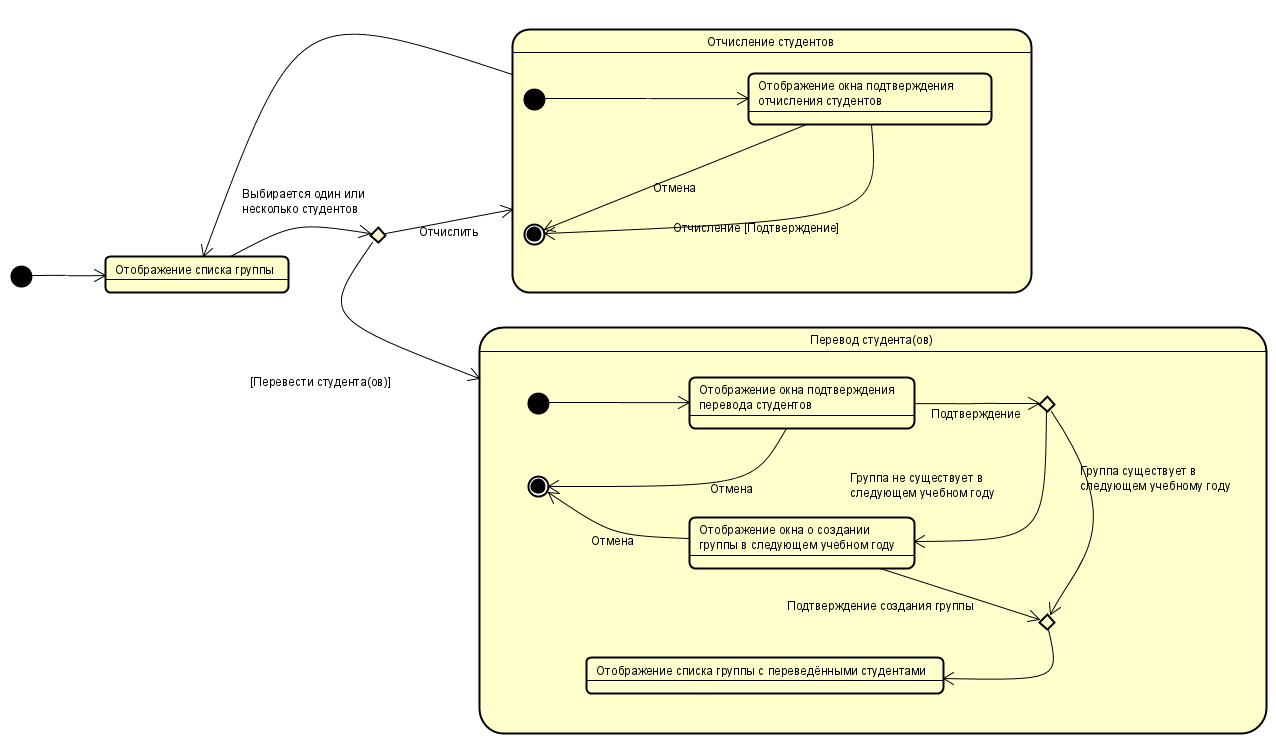


Рисунок 12 – Диаграмма состояния “Перевод / отчисление отдельного студента”

# 3 Модель данных

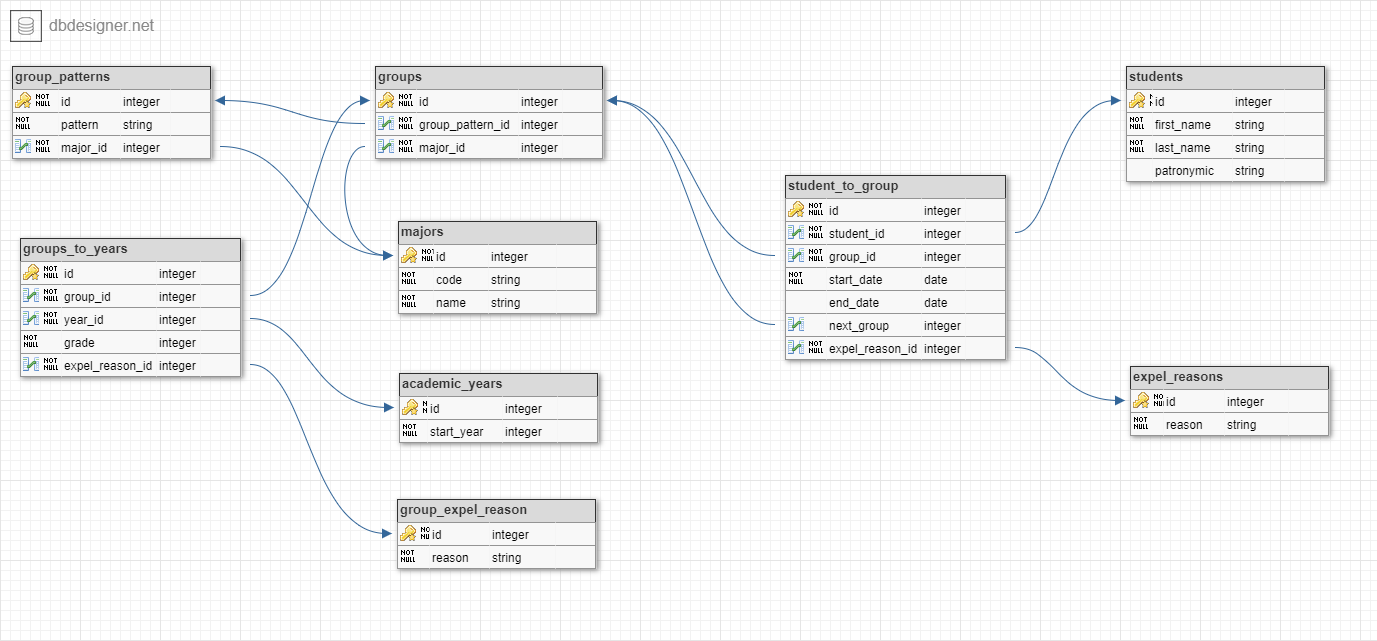


Рисунок 13 – Физическая модель данных

# 4 Используемые инструменты

* + - 1. Языки программирования:
* PHP 7.3 - серверная часть приложения.
* JavaScript - клиентская часть приложения.

Фреймворки:

* Laravel - серверная часть приложения.

Библиотеки готового кода:

* JQuery - упрощенное взаимодействие с DOM в клиентской части.
* Bootstrap - готовые стили клиентской части

# 5 Маршрутизация

Таблица 1. Маршрутизация

|  |  |  |
| --- | --- | --- |
| Тип запроса | Путь | Описание |
| GET | /groups/{yearId} | Страница с группами |
| GET | /groups/{yearId}/{id} | Страница группы с указанным  идентификатором |
| GET | groups/{yearId}/{id}/student/add/ | Страница создания добавления студента в группу |
| GET | /students | Страница списка студентов |
| GET | /students/{id} | Страница конкретного студента с указанным идентификатором |
| GET | /groups/{yearId}/new | Страница создания группы в заданном учебном году |
| Rest API (/api) | | |
| GET | /groups/{yearId} | Получить группы для заданного года |
| GET | /groups/{yearId}/major/{majorId} | Получить группы для заданного года и заданного направления |
| GET | /groups/{yearId}/id/{id} | Получить группу для заданного года с заданным ID |
| POST | /groups/{yearId} | Создать группу в заданном учебном году |
| PUT | /groups/{yearId}/{id} | Изменить группу для заданного года с заданным ID |
| DELETE | /groups/{yearId}/{id} | Удалить группу для заданного года с заданным ID |
| PUT | /groups/{yearId}/{id}/nextGrade | Первести группу на следующий год |
| GET | /years | Получить все учебные года |
| GET | /years/{id} | Получить учебный год по указанному id |
| POST | /years | Создать новый учебный год |
| DELETE | /years/{id} | Удалить учебный год |
| GET | /students | Получит всех студентов |
| GET | /students/{id} | Получить студента с заданным id |
| GET | /students/{id}/history | Получить историю обучения заданного студента |
| POST | /students | Создать студента |
| PUT | /students/{id} | Обновить данные студента с заданным id |
| PUT | /students/{id}/transfer | Перевести студента в другую группу |
| DELETE | /students/{id} | Удалить студента с заданным id |
| GET | /patterns | Получить шаблоны групп |
| GET | /patterns/{id} | Получить шаблон по заданному ID |
| GET | /patterns/major/{majorId} | Получить шаблоны для заданного направления |
| POST | /patterns/major/{majorId} | Создать шаблон для заданного направления |
| PUT | /patterns/{id} | Обновить шаблон по заданному ID |
| DELETE | /patterns/{id} | Удалить шаблон по заданному ID |
| GET | /majors | Получить список направлений |
| GET | /majors/{id} | Получить направление по заданному id |
| POST | /majors | Создать направление |
| PUT | /majors/{id} | Обновить направление с заданным id |
| DELETE | /majors/{id} | Удалить направление с заданным id |

# 6 Репозиторий кода

https://github.com/MDKP2020/Noskin

# 7 Тестирование

Использовалось ручное тестирование. Список найденных ошибок не вёлся.

# 8 Полученное программное решение

Программное решение развёрнуто удаленно. Ссылка: http://noskin.test1.seschool.ru/

# 9 Вклад участников команды

Таблица 2. Вклад участников команды

|  |  |  |  |
| --- | --- | --- | --- |
|  | Носкин Виктор | Дудкин Даниил | Крымова Мария |
| Модель бизнес-процессов  (сценарии) | 0 | 0 | 1 |
| Макеты экранных форм | 0 | 0 | 1 |
| Модель данных | 0 | 1 | 0 |
| Миграция БД | 0 | 1 | 0 |
| Верстка | 0,6 | 0,3 | 0,1 |
| Реализация представлений | 0,7 | 0,3 | 0 |
| Реализация контроллеров | 0,7 | 0,3 | 0 |
| Реализация моделей | 0,6 | 0,4 | 0 |
| Тестирование (в примечании указать что и способ) | 0,5 (ручное) | 0,5 (ручное) | 0 |
| Отладка, если разные исполнители реализовывали и отлаживали один и тот же артефакт | 0,5 | 0,5 | 0 |
| Мердж в общую ветку | 1 | 0 | 0 |
| Настройка и сопровождение удаленного сервера / приложения | 1 | 0 | 0 |
| Управление проектом / задачами | 1 | 0 | 0 |
| Документация + презентация + диаграммы | 1 | 0 | 0 |

# 10 Используемые источники

1. Bootstrap [Электронный ресурс] : офиц. сайт. – 2021. – Режим доступа : https://getbootstrap.com/docs
2. Laravel [Электронный ресурс] : офиц. сайт. – 2021. – Режим доступа : https://laravel.com
3. PHP [Электронный ресурс] : офиц. сайт. – 2021. – Режим доступа : https://php.net/manual
4. Макконнелл С. Совершенный код. Мастер-класс / Пер. с англ. — М. : Издательство «Русская редакция», 2010. — 896 стр.